PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 7 :		(11) Numéro de publication internationale:
H04L 9/06, 9/30	A1	WO 00/46953
		(43) Date de publication internationale: 10 août 2000 (10.08.00)

(21) Numéro de la demande internationale: PCT/FR00/00258

(22) Date de dépôt international: 3 février 2000 (03.02.00)

(30) Données relatives à la priorité:
99/01289 4 février 1999 (04.02.99) FR

(71) Déposant: BULL CP8 [FR/FR]; 68, route de Versailles, Boîte postale 45, F–78430 Louveciennes (FR).

(72) Inventeurs: PATARIN, Jacques; 11, rue Amédée Dailly, F–78220 Viroflay (FR). GOUBIN, Louis; 3, rue Brown–Séquard, F–75015 Paris (FR).

(74) Mandataire: CORLU, Bernard; Bull S.A., PC58D20, 68, route de Versailles, F–78434 Louveciennes Cedex (FR).

(81) Etats désignés: JP, brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

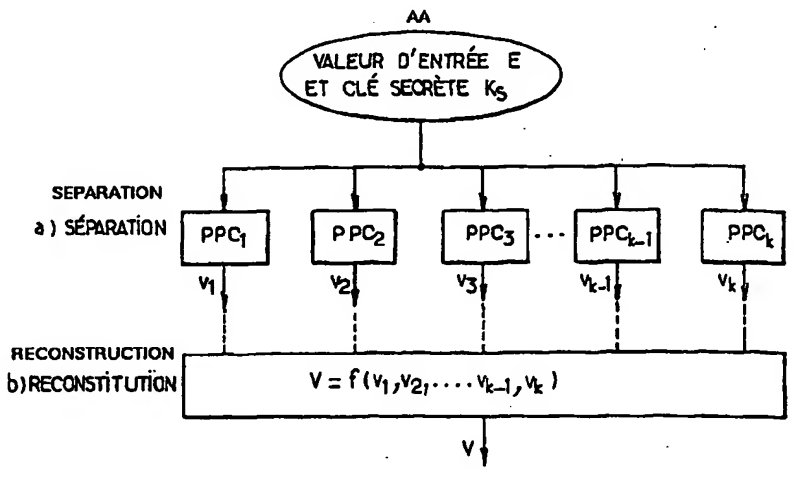
Publiée
Avec rapport de recherche internationale.

(54) Title: METHOD FOR PROTECTING AN ELECTRONIC CRYPTOGRAPHIC SET WITH SECRET KEY AGAINST CRYPTANALYTICAL ATTACK

(54) Titre: PROCEDE DE SECURISATION D'UN ENSEMBLE ELECTRONIQUE DE CRYPTOGRAPHIE A CLE SECRETE CONTRE LES ATTAQUES PAR ANALYSE PHYSIQUE

(57) Abstract

The invention concerns a method for protecting an electronic cryptographic set with secret key against cryptanalytical attack, which consists in dividing a) the standard cryptographic computation process into several separate parallel processing stages using partial intermediate results distinct from those of standard computation; and b) reconstructing the final value from the separate intermediate partial results. The invention is useful for electronic sets such as incorporated systems.



AA ... INPUT VALUE E AND SECRET KEY Ks

(57) Abrégé

L'invention concerne un procédé de sécurisation d'un ensemble électronique de calcul cryptographique à clé secrète contre les attaques physiques. L'on divise a) le processus de calcul cryptographique classique en plusieurs parties de processus distinctes en parallèle mettant en oeuvre des résultats partiels intermédiaires distincts de ceux du calcul classique et b) on reconstitue la valeur finale du calcul cryptographique classique à partir des résultats partiels intermédiaires distincts. Application d'ensembles électroniques tels que les systèmes embarqués.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	**ES**	Espagne	**LS**	Lesotho	**SI**	Slovénie
AM	Arménie	**FI**	Finlande	**LT**	Lituanie	**SK**	Slovaquie
AT	Autriche	**FR**	France	**LU**	Luxembourg	**SN**	Sénégal
AU	Australie	**GA**	Gabon	**LV**	Lettonie	**SZ**	Swaziland
AZ	Azerbaïdjan	**GB**	Royaume-Uni	**MC**	Monaco	**TD**	Tchad
BA	Bosnie-Herzégovine	**GE**	Géorgie	**MD**	République de Moldova	**TG**	Togo
BB	Barbade	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tadjikistan
BE	Belgique	**GN**	Guinée	**MK**	Ex-République yougoslave de Macédoine	**TM**	Turkménistan
BF	Burkina Faso	**GR**	Grèce			**TR**	Turquie
BG	Bulgarie	**HU**	Hongrie	**ML**	Mali	**TT**	Trinité-et-Tobago
BJ	Bénin	**IE**	Irlande	**MN**	Mongolie	**UA**	Ukraine
BR	Brésil	**IL**	Israël	**MR**	Mauritanie	**UG**	Ouganda
BY	Bélarus	**IS**	Islande	**MW**	Malawi	**US**	Etats-Unis d'Amérique
CA	Canada	**IT**	Italie	**MX**	Mexique	**UZ**	Ouzbékistan
CF	République centrafricaine	**JP**	Japon	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Pays-Bas	**YU**	Yougoslavie
CH	Suisse	**KG**	Kirghizistan	**NO**	Norvège	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	République populaire démocratique de Corée	**NZ**	Nouvelle-Zélande		
CM	Cameroun			**PL**	Pologne		
CN	Chine	**KR**	République de Corée	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Roumanie		
CZ	République tchèque	**LC**	Sainte-Lucie	**RU**	Fédération de Russie		
DE	Allemagne	**LI**	Liechtenstein	**SD**	Soudan		
DK	Danemark	**LK**	Sri Lanka	**SE**	Suède		
EE	Estonie	**LR**	Libéria	**SG**	Singapour		

PROCEDE DE SECURISATION D'UN ENSEMBLE ELECTRONIQUE DE CRYPTOGRAPHIE A CLE SECRETE CONTRE LES ATTAQUES PAR ANALYSE PHYSIQUE

La présente invention concerne un procédé de sécurisation d'un ensemble électronique mettant en œuvre un algorithme cryptographique qui utilise une clé secrète. Plus précisément, le procédé vise à réaliser une version de l'algorithme qui ne soit pas vulnérable face à un certain type d'attaques physiques - dites *Differential Power Analysis* ou *High-Order Differential Power Analysis* - qui cherchent à obtenir des informations sur la clé secrète à partir de l'étude de la consommation électrique de l'ensemble électronique au cours de l'exécution du calcul.

Les algorithmes cryptographiques considérés ici utilisent une clé secrète pour calculer une information de sortie en fonction d'une information d'entrée ; il peut s'agir d'une opération de chiffrement, de déchiffrement ou de signature ou de vérification de signature, ou d'authentification ou de non-répudiation. Ils sont construits de manière à ce qu'un attaquant, connaissant les entrées et les sorties, ne puisse en pratique déduire aucune information sur la clé secrète elle-même.

On s'intéresse donc à une classe plus large que celle traditionnellement désignée par l'expression *algorithmes à clé secrète* ou *algorithmes symétriques*. En particulier, tout ce qui est décrit dans la présente demande de brevet s'applique également aux algorithmes dits *à clé publique* ou *algorithmes asymétriques*, qui comportent en fait deux clés : l'une publique, et l'autre, privée, non divulguée, cette dernière étant celle visée par les attaques décrites ci-dessous.

Les attaques de type Analyse de Puissance Electrique, *Power Analysis* en langage anglo-saxon, développées par Paul Kocher et *Cryptographic Research* (Confer document *Introduction to Differential Power Analysis and related Attacks* by Paul Kocher, Joshua Jaffe, and Benjamin Jun, Cryptography Research, 870 Market St., Suite 1008, San Francisco, CA 94102, édition du document HTML à l'adresse URL : http://www.cryptography.com/dpa/technical/index.html, introduit dans la présente demande à titre de référence) partent de la constatation qu'en réalité l'attaquant peut acquérir des informations, autres que la simple donnée des entrées et des sorties, lors de l'exécution du calcul, comme par exemple la consommation électrique du microcontrôleur ou le rayonnement électromagnétique émis par le circuit.

L'analyse d'énergie électrique différentielle, *Differential Power Analysis* en langage anglo-saxon, en abrégé DPA, est une attaque permettant d'obtenir des informations sur la clé secrète contenue dans l'ensemble électronique, en effectuant une analyse statistique des enregistrements de consommation électrique effectués sur un grand nombre de calculs avec cette même clé.

On considère, à titre d'exemple non limitatif, le cas de l'algorithme DES (*Data Encryption Standard*), dont on peut trouver une description dans l'un des documents suivants :

- FIPS PUB 46-2, *Data Encryption Standard*, 1994;
- FIPS PUB 74, *Guidelines for Implementing and Using the NBS Data Encryption Standard*, 1981;
- ANSI X3.92, *American National Standard, Data Encryption Algorithm*, 1981;

- ISO/IEC 8731:1987, *Banking - Approved Algorithms for Message Authentication - Part 1: Data Encryption Algorithm (DEA)*.

ou encore dans l'ouvrage suivant :

- Bruce Schneier, *Applied Cryptography*, 2ème édition, John Wiley & Sons, 1996, page 270.

Les documents précités sont introduits dans la présente demande à titre de référence.

L'algorithme DES se déroule en 16 étapes appelées *tours*, confer figure 1a. Dans chacun des 16 tours, une transformation *F* est effectuée sur 32 bits. Cette transformation *F* utilise huit transformations non linéaires de 6 bits sur 4 bits, qui sont codées chacune dans une table appelée *boîte-S*, confer figure 1b, où les boîtes S sont notées S_1, S_2, ..., S_8.

L'attaque DPA sur le DES peut être mise en œuvre de la manière suivante :

- 1ère étape : On fait des mesures de consommation sur le premier tour, ceci pour 1000 calculs de DES. On note *E[1], ..., E[1000]* les valeurs d'entrée de ces 1000 calculs. On note *C[1], ..., C[1000]* les 1000 courbes correspondantes de consommation électrique mesurées lors de ces calculs. On calcule également la courbe *moyenne CM* des 1000 courbes de consommation.
- 2ème étape : On s'intéresse, par exemple, au premier bit de sortie de la première boîte-S lors du premier tour. Notons *b* la valeur de ce bit. Il est facile de voir que *b* ne dépend que de 6 bits de la clé secrète. L'attaquant fait une hypothèse sur les 6 bits concernés. Il calcule - à partir de ces 6 bits et des *E[i]* - les valeurs théoriques attendues pour *b*. Cela permet de séparer les

1000 entrées *E[1], ..., E[1000]* en deux catégories : celles qui donnent *b=0*, et celles qui donnent *b=1*.

- 3ème étape : On calcule maintenant la moyenne *CM'* des courbes correspondant à des entrées de la première catégorie, c'est-à-dire pour lesquelles *b=0*. Si *CM* et *CM'* présentent une différence notable, on considère que les valeurs retenues pour les 6 bits de clé étaient les bonnes. Si *CM* et *CM'* ne présentent pas de différence sensible, au sens statistique, c'est-à-dire pas de différence nettement supérieure à l'écart type du bruit mesuré, on recommence la 2ème étape avec un autre choix pour les 6 bits.
- 4ème étape : On répète les étapes 2 et 3 avec un bit cible *b* issu de la deuxième boîte-S, puis de la troisième boîte-S, ..., jusqu'à la huitième boîte-S. On obtient donc finalement 48 bits de la clé secrète.
- 5ème étape : Les 8 bits restants peuvent être trouvés par recherche exhaustive.

Cette attaque ne nécessite aucune connaissance sur la consommation électrique individuelle de chaque instruction, ni sur la position dans le temps de chacune de ces instructions. Elle s'applique de la même manière si on suppose que l'attaquant connaît des sorties de l'algorithme et les courbes de consommation correspondantes. Elle repose uniquement sur l'hypothèse fondamentale selon laquelle :

Hypothèse fondamentale : Il existe une variable intermédiaire, apparaissant dans le cours du calcul de l'algorithme, telle que la connaissance de quelques bits de clé, en pratique moins de 32 bits, permet de décider si deux entrées, respectivement deux sorties, donnent ou non la même valeur pour cette variable.

Tous les algorithmes utilisant des boîtes-S, tels le DES, sont potentiellement vulnérables à la DPA, car les modes de réalisation usuels restent en général dans le cadre de l'hypothèse mentionnée ci-dessus.

Les attaques dites par analyse d'énergie électrique de haut niveau, *High-Order Differential Power Analysis* en langage anglo-saxon, en abrégé HO-DPA, sont une généralisation de l'attaque DPA décrite précédemment. Elles peuvent utiliser plusieurs sources d'information différentes, outre la consommation elles peuvent mettre en jeu les mesures de rayonnement électromagnétique, de température, etc. et mettre en œuvre des traitements statistiques plus sophistiqués que la simple notion de moyenne, des variables intermédiaires (généralisant le bit *b* défini ci-dessus) moins élémentaires. Néanmoins, elles reposent exactement sur la même hypothèse fondamentale que la DPA.

Le procédé, objet de la présente invention, a pour objet la suppression des risques d'attaques DPA ou HO-DPA d'ensembles ou systèmes électroniques de cryptographie à clé secrète ou privée.

Un autre objet de la présente invention est en conséquence une modification du processus de calcul cryptographique mis en œuvre par les systèmes électroniques de cryptographie protégés de manière que l'hypothèse fondamentale précitée ne soit plus vérifiée, à savoir qu'aucune variable intermédiaire ne dépend de la consommation d'un sous-ensemble aisément accessible de la clé secrète ou privée, les attaques de type DPA ou HO-DPA étant ainsi rendues inopérantes.

Le procédé de sécurisation d'un ensemble électronique mettant en œuvre un processus de calcul cryptographique classique qui utilise une clé secrète,

objet de la présente invention, est remarquable en ce que l'on divise le processus de calcul cryptographique en plusieurs parties de calcul distinctes conduites parallèlement et mettant en œuvre des résultats partiels intermédiaires distincts de ceux du calcul cryptographique classique et que l'on reconstitue la valeur finale, obtenue par le calcul classique en l'absence de division, à partir des résultats partiels intermédiaires distincts. Par processus de calcul cryptographique classique, on entend tout processus de calcul séquentiel ou successif permettant d'obtenir des valeurs chiffrées, déchiffrées, des valeurs de signature, de vérification de signature, d'authentification et de non-répudiation. Un tel procédé permet d'inhiber les attaques de type DPA ou HO-DPA contre les ensembles ou systèmes embarqués munis de fonctions de calcul cryptographique tels que les cartes à microcalculateurs dédiées à des fonctions de monétique électronique, carte bancaire, carte de contrôle d'accès ou analogues.

Il sera mieux compris à la lecture de la description et à l'observation des dessins ci-après dans lesquels, outre les figures 1a et 1b relatives à l'art antérieur en référence au processus de chiffrement/ déchiffrement DES pour *"Data Encryption System"* en langage anglo-saxon :

- la figure 2 représente un organigramme général illustratif du procédé objet de l'invention ;
- la figure 3a représente, de manière illustrative, un mode de mise en œuvre non limitatif du procédé objet de la présente invention ;
- la figure 3b représente, à titre d'exemple illustratif, un organigramme d'une mise en œuvre particulière du procédé objet de l'invention appliqué à

une transformation non linéaire utilisée dans un processus de calcul cryptographique classique tel que le DES ;

- la figure 3c représente une variante de mise en œuvre du procédé objet de l'invention tel qu'illustré en figure 2 ;

- la figure 3d représente, à titre d'exemple illustratif, un organigramme d'une autre mise en œuvre particulière du procédé objet de l'invention, à partir d'une transformation bijective secrète, appliqué à une transformation non linéaire utilisée dans un processus de calcul cryptographique classique tel que le DES ;

- la figure 3e représente, à titre d'exemple illustratif, un organigramme d'une autre mise en œuvre particulière du procédé objet de l'invention, à partir de fonctions polynomiales, appliqué à une transformation non linéaire utilisée dans un processus de calcul cryptographique classique tel que le DES.

Une description plus détaillée du procédé de sécurisation d'un ensemble électronique mettant en œuvre un processus de calcul cryptographique classique qui utilise une clé secrète, objet de la présente invention, sera maintenant donnée en liaison avec les figures précitées.

D'une manière générale, le procédé objet de la présente invention consiste, pour un processus de calcul cryptographique classique qui utilise une clé secrète Ks, ou privée, à modifier le processus de calcul cryptographique de manière que l'hypothèse fondamentale précitée ne soit plus vérifiée, aucune variable intermédiaire calculée ne dépendant plus, conformément au procédé objet de la présente invention, de la connaissance d'un sous-ensemble aisément accessible de la clé secrète.

Dans ce but, et conformément au procédé objet de la présente invention tel que représenté en figure 2, a) on divise le processus de calcul cryptographique classique en plusieurs parties de processus de calcul PPC_1 à PPC_k distinctes conduites parallèlement, puis b) on reconstitue la valeur finale V correspondant à celle obtenue par le calcul cryptographique classique, en l'absence de division, à partir des résultats partiels intermédiaires distincts v_1 à v_k obtenus par la mise en œuvre des parties de processus de calcul distinctes PPC_1 à PPC_k précitées.

Ainsi, les parties de processus de calcul sont indépendantes mais les variables ou résultats intermédiaires partiels sont liés.

On réalise cette division en remplaçant chaque variable *v* intermédiaire, intervenant dans le cours du calcul et dépendant des données d'entrée (ou de sortie), par *k* variables v_1, v_2, ..., v_k, telles que v_1, v_2, ..., et v_k permettent, au besoin, de reconstituer v. Plus précisément, cela signifie qu'il existe une fonction *f* telle que $v=f(v_1, v_2, \ldots, v_k)$. On indique en outre que *f* satisfait, de préférence, la condition suivante :

Condition n°1 :

Soit *i* un indice compris, au sens large, entre *1* et *k*. La connaissance d'une valeur *v* ne permet jamais en pratique de déduire des informations sur l'ensemble des valeurs v_i telles qu'il existe un *(k-1)-uplet* $(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k)$ satisfaisant l'équation $f(v_1, \ldots, v_k)=v$;

Exemple n°1 :

Si on prend comme fonction $f(v_1, \ldots, v_k)= v_1 \oplus v_2 \oplus \ldots \oplus v_k$, où $\oplus$ désigne la fonction "OU-exclusif" bit à bit, la condition n°1 est réalisée de manière manifeste, car, pour tout indice *i* compris entre *1* et *k*, l'ensemble

considéré des valeurs v_i contient toutes les valeurs possibles, et ne dépend donc pas de v.

Exemple n°2 :

Si on considère une variable v à valeurs dans le groupe multiplicatif de Z/nZ, c'est-à-dire l'ensemble des entiers modulo n qui possèdent un inverse, également modulo n, on peut prendre comme fonction $f(v_1, \ldots, v_k) = v_1 \cdot v_2 \cdot \ldots \cdot v_k \bmod n$, où les nouvelles variables $v_1, v_2, \ldots, v_k$ sont également à valeurs dans le groupe multiplicatif de Z/nZ. La condition n°1 est aussi réalisée de manière manifeste, car, pour tout indice i compris entre 1 et k, l'ensemble considéré des valeurs v_i contient toutes les valeurs possibles, et ne dépend donc pas de v.

Conformément à un aspect remarquable du procédé objet de l'invention, on effectue alors une "*traduction*" de l'algorithme en remplaçant chaque variable intermédiaire v dépendant des données d'entrée (ou de sortie) par les k variables $v_1, v_2, \ldots, v_k$. Pour garantir la sécurité maximale de l'algorithme modifié sous sa nouvelle forme, on impose la condition supplémentaire suivante sur la fonction f :

Condition n°2 :

La fonction f est telle que les transformations à effectuer sur $v_1, v_2, \ldots,$ ou v_k au cours du calcul, à la place des transformations effectuées habituellement sur v, peuvent être exécutées sans avoir à recalculer v.

Premier exemple : le DES

Un premier exemple concernant la sécurisation du DES sera décrit en liaison avec la figure 3a.

Dans cet exemple, on considère le cas particulier du DES. On choisit ici une division de chaque variable v intermédiaire, intervenant dans le cours du calcul et dépendant des données d'entrée, ou de sortie, en deux variables v_1 et v_2, c'est-à-dire que l'on prend $k=2$. On considère la fonction $f(v_1, v_2) = v = v_1 \oplus v_2$ de l'exemple n°1 ci-dessus, qui satisfait par construction la condition n°1. Par construction de l'algorithme, on constate facilement que les transformations qu'il effectue sur v peuvent toujours entrer dans l'une des cinq catégories suivantes :

- permutation des bits de v ;
- expansion des bits de v ;
- OU-exclusif de v avec une autre variable v' du même type ;
- OU-exclusif de v avec une variable c dépendant uniquement de la clé ou d'une sous-clé ;
- transformation non linéaire de v par une boîte-S.

Les deux premières catégories correspondent à des transformations linéaires sur les bits de la variable v. Pour celles-ci, la condition n°2 est donc très facile à vérifier : il suffit, à la place de la transformation effectuée habituellement sur v, d'effectuer la permutation ou l'expansion sur v_1, puis sur v_2, et la relation $f(v_1, v_2) = v$ qui était vraie avant la transformation reste vraie également après.

De même, dans le troisième cas, il suffit de remplacer le calcul de $v'' = v \oplus v'$ par celui de $v''_1 = v_1 \oplus v'_1$ et de $v''_2 = v_2 \oplus v'_2$. Les relations $f(v_1, v_2) = v$ et $f(v'_1, v'_2) = v'$ donnent bien $f(v''_1, v''_2) = v''$, et la condition n°2 est encore vérifiée.

En ce qui concerne le OU-exclusif de v avec une variable c dépendant uniquement de la clé ou d'une sous-clé, la condition n°2 est aussi très facile à satisfaire : il suffit de remplacer le calcul de $v \oplus c$ par $v_1 \oplus c$, ou $v_2 \oplus c$, ce qui assure la condition n°2.

Enfin, à la place de la transformation non-linéaire $v'=S(v)$, donnée sous la forme d'une boîte-S, qui dans cet exemple admet des entrées de 6 bits et donne des sorties de 4 bits, on réalise la transformation $(v'_1,v'_2)=S'(v_1,v_2)$ au moyen de deux nouvelles boîtes S, chacune étant cette fois de 12 bits sur 4 bits. Pour garantir l'égalité $f(v'_1, v'_2) = v'$, il suffit de choisir :

$$(v'_1, v'_2) = S'(v_1, v_2) = (A(v_1, v_2), S(v_1 \oplus v_2) \oplus A(v_1, v_2))$$

c'est-à-dire $v'_1 = A(v_1, v_2)$ et $v'_2 = S(v_1 \oplus v_2) \oplus A(v_1, v_2)$

où A désigne une transformation *aléatoire* et *secrète* de 12 bits vers 4 bits. La première (nouvelle) boîte-S correspond à la table de la transformation $(v_1, v_2) \rightarrow A(v_1, v_2)$ qui à (v_1, v_2) associe $A(v_1, v_2)$ et la seconde (nouvelle) boîte-S correspond à la table de la transformation $(v_1, v_2) \rightarrow S(v_1 \oplus v_2) \oplus A(v_1, v_2)$ qui à (v_1, v_2) associe $S(v_1 \oplus v_2) \oplus A(v_1, v_2)$. La présence de la fonction aléatoire A permet de garantir la condition n°1. L'utilisation de tables permet par ailleurs d'éviter

d'avoir à calculer $v_1 \oplus v_2$ et, par là, permet de satisfaire la condition n°2.

Les tables de transformation ou de conversion peuvent être mémorisées dans une mémoire ROM de la carte à microcalculateur lorsque l'ensemble électronique est constitué par une carte à microcalculateur.

Ainsi, pour une étape de calcul du type transformation non linéaire mise en œuvre par un processus de calcul cryptographique classique tel que le DES, la division, ainsi que représenté en figure 3b, peut être effectuée en k parties. Pour un processus de calcul cryptographique classique utilisant des transformations non linéaires de m bits sur n bits, décrites par des tables de conversion dans lesquelles les n bits de sortie de la transformation sont lus à une adresse fonction des m bits d'entrée, on remplace chaque transformation non linéaire appliquée à une variable intermédiaire jouant le rôle de variable d'entrée E du processus de calcul cryptographique classique, en l'absence de division, par une transformation non linéaire partielle de km bits sur kn bits appliquée à l'ensemble des variables intermédiaires partielles v_1 à v_k. Selon un aspect particulièrement remarquable du procédé objet de l'invention, cette transformation non linéaire partielle est décrite et réalisée par k tables de conversion partielle dans lesquelles les n bits de sortie v'_1 ou v'_2 … ou v'_k de la transformation sont lus à une adresse fonction des km bits d'entrée.

Dans le premier exemple précité et en relation avec la figure 3b, on indique que k=2, n=4 et m=6.

Variante n°1 :

Pour des raisons d'encombrement de la ROM, on peut tout à fait utiliser la même fonction aléatoire *A* pour chacune des huit boîtes S de la description classique du DES, ce qui permet de n'avoir que neuf nouvelles boîtes-S à stocker au lieu de seize.

Une variante, variante n°2, sera décrite en liaison avec la figure 3c.

Variante n°2 :

Afin de réduire la taille de la ROM nécessaire pour stocker les boîtes S, on peut également utiliser la méthode suivante : à la place de chaque transformation non-linéaire $v'=S(v)$ de l'implémentation initiale, donnée sous la forme d'une boîte-S (qui dans l'exemple du DES admet des entrées de 6 bits et donne des sorties de 4 bits), on réalise la transformation $(v'_1,v'_2)=S'(v_1,v_2)$ au moyen de deux boîtes S, chacune étant cette fois de 6 bits sur 4 bits. La mise en œuvre initiale du calcul de $v'=S(v)$ est remplacée par les deux calculs successifs suivants :

- $v_0=\varphi(v_1 \oplus v_2)$
- $(v'_1, v'_2) = S'(v_1, v_2) = (A(v_0), S(\varphi^{-1}(v_0)) \oplus A(v_0))$

c'est-à-dire $v'_1 = A(v_0)$, $v'_2 = S(\varphi^{-1}(v_0)) \oplus A(v_0)$

où φ est une fonction bijective et secrète de 6 bits sur 6 bits, et où *A* désigne une transformation *aléatoire* et *secrète* de 6 bits vers 4 bits. La première (nouvelle) boîte-S correspond à la table de la transformation $v_0 \rightarrow A(v_0)$ qui à v_0 associe $A(v_0)$ et la seconde (nouvelle) boîte-S correspond à la table de la transformation $v_0 \rightarrow S($

$\varphi^{-1}(v_0)) \oplus A(v_0)$ qui à v_0 associe $S(\varphi^{-1}(v_0)) \oplus A(v_0)$. Par construction, on a toujours l'égalité $f(v'_1, v'_2) = v'$. La présence de la fonction aléatoire A permet de garantir la condition n°1. L'utilisation de tables permet d'éviter d'avoir à calculer $\varphi^{-1}(v_0) = v_1 \oplus v_2$.

Sur la figure 3d, on a représenté une étape de calcul correspondante, de type transformation non linéaire mise en œuvre dans le cadre du processus de calcul cryptographique classique tel que le DES, tel que modifié conformément au procédé objet de l'invention selon la Variante n°2. Outre la division en k parties appliquée à la variable d'entrée E, pour les transformations non linéaires de m bits sur n bits, décrites par des tables de conversion dans lesquelles les n bits de sortie sont lus à une adresse fonction des m bits d'entrée, on remplace chaque transformation non linéaire appliquée à une variable intermédiaire, jouant le rôle de variable d'entrée E, du processus de calcul classique par une transformation non linéaire partielle de km bits sur kn bits appliquée sur l'ensemble des variables intermédiaires partielles v_1 à v_k. Cette transformation non linéaire partielle est décrite et réalisée par k tables de conversion, chacune des entrées des tables de conversion recevant une valeur obtenue par application d'une fonction bijective secrète φ_j à la fonction $f(v_1, \ldots, v_k)$ des variables intermédiaires partielles suivant la relation $\varphi_j \circ f(v_1, \ldots, v_k)$, avec $j \in [1,k]$.

Selon un aspect particulièrement remarquable du procédé, objet de la présente invention, l'application précitée $\varphi_j \circ f(v_1, \ldots, v_k)$ est effectuée par évaluation directe d'une valeur résultante, laquelle, appliquée à

l'entrée de la table de conversion correspondante 1 à k, permet de lire n bits de sortie de la transformation v'_1 ou v'_2 ou ...v'_k à une adresse qui est fonction de ces m bits d'entrée.

De même que dans le premier exemple précité, et en relation avec la figure 3d, on indique que pour la Variante n°2, k=2, m=6 et n=4.

En outre, et dans une version simplifiée, les fonctions bijectives φ_1 à φ_k sont identiques.

Pour que la condition n°2 soit satisfaite, il reste à choisir la transformation bijective φ ou des fonctions bijectives φ_1 à φ_k de telle sorte que le calcul de $v_0=\varphi(v_1 \oplus v_2)$ puisse se faire sans avoir à recalculer $v_1 \oplus v_2$. Deux exemples de choix pour la fonction φ sont donnés ci-après :

Exemple 1 : Une bijection φ linéaire

On choisit pour φ une fonction linéaire secrète et bijective de 6 bits sur 6 bits. Dans le cadre d'un tel choix, on considère l'ensemble des valeurs sur 6 bits comme un espace vectoriel de dimension 6 sur le corps fini F_2 à deux éléments. En pratique, choisir φ revient à choisir une matrice aléatoire et inversible de taille 6x6 dont les coefficients valent 0 ou 1. Avec ce choix de φ, il est facile de voir que la condition n°2 est satisfaite. En effet - pour calculer $\varphi(v_1 \oplus v_2)$ - il suffit de calculer $\varphi(v_1)$, puis $\varphi(v_2)$, et enfin de calculer le "OU-exclusif" des deux résultats obtenus.

Par exemple, la matrice $\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$ est inversible. Il lui correspond la bijection linéaire φ de 6 bits sur 6 bits définie par :

$\varphi(u_1, u_2, u_3, u_4, u_5, u_6) = (u_1 \oplus u_2 \oplus u_4, u_1 \oplus u_2 \oplus u_4 \oplus u_6, u_2 \oplus u_3 \oplus u_5, u_1 \oplus u_2 \oplus u_3 \oplus u_5, u_2 \oplus u_3 \oplus u_4 \oplus u_5, u_3 \oplus u_4 \oplus u_6)$

Si on note $v_1 = (v_{1,1}, v_{1,2}, v_{1,3}, v_{1,4}, v_{1,5}, v_{1,6})$ et $v_2 = (v_{2,1}, v_{2,2}, v_{2,3}, v_{2,4}, v_{2,5}, v_{2,6})$, pour calculer $\varphi(v_1 \oplus v_2)$, on calcule successivement :

- $\varphi(v_1) = (v_{1,1} \oplus v_{1,2} \oplus v_{1,4}, v_{1,1} \oplus v_{1,2} \oplus v_{1,4} \oplus v_{1,6}, v_{1,2} \oplus v_{1,3} \oplus v_{1,5}, v_{1,1} \oplus v_{1,2} \oplus v_{1,3} \oplus v_{1,5}, v_{1,2} \oplus v_{1,3} \oplus v_{1,4} \oplus v_{1,5}, v_{1,3} \oplus v_{1,4} \oplus v_{1,6})$;
- $\varphi(v_2) = (v_{2,1} \oplus v_{2,2} \oplus v_{2,4}, v_{2,1} \oplus v_{2,2} \oplus v_{2,4} \oplus v_{2,6}, v_{2,2} \oplus v_{2,3} \oplus v_{2,5}, v_{2,1} \oplus v_{2,2} \oplus v_{2,3} \oplus v_{2,5}, v_{2,2} \oplus v_{2,3} \oplus v_{2,4} \oplus v_{2,5}, v_{2,3} \oplus v_{2,4} \oplus v_{2,6})$.

Puis on calcule le "*OU-exclusif*" des deux résultats obtenus.

Exemple 2 : Une bijection φ quadratique

On choisit pour φ une fonction quadratique secrète et bijective de 6 bits sur 6 bits. Le terme "*quadratique*" signifie ici que chaque bit de valeur de sortie de la fonction φ est donné par une fonction polynomiale de *degré deux* des 6 bits d'entrée, qui sont identifiés à 6 éléments du corps fini F_2. En pratique, on peut choisir la fonction

φ définie par la formule $\varphi(x)=t(s(x)^5)$, où s est une application linéaire secrète et bijective de $(F_2)^6$ sur L, t est une application linéaire secrète et bijective de L sur $(F_2)^6$, et où L désigne une extension algébrique de degré 6 du corps fini F_2. Le caractère bijectif de cette fonction φ résulte du fait que $a \rightarrow a^5$ est une bijection sur l'extension L (dont l'inverse est $b \rightarrow b^{38}$). Pour établir que la condition n°2 est encore satisfaite, il suffit de remarquer que l'on peut écrire :

$$\varphi(v_1 \oplus v_2) = \psi(v_1, v_1) \oplus \psi(v_1, v_2) \oplus \psi(v_2, v_1) \oplus \psi(v_2, v_2)$$

où la fonction ψ est définie par : $\psi(x, y)=t(s(x)^4 \cdot s(y))$.

Par exemple, si on identifie L à $F_2[X]/(X^6+X+1)$, et si on prend s et t de matrices respectives

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

par rapport à la base $(1, X, X^2, X^3, X^4, X^5)$ de L sur F_2 et à la base canonique de $(F_2)^6$ sur F_2, on obtient la bijection quadratique φ de 6 bits sur 6 bits suivante :

$$\varphi(u_1, u_2, u_3, u_4, u_5, u_6) =$$
$$(u_2u_5 \oplus u_1u_4 \oplus u_4 \oplus u_6 \oplus u_6u_2 \oplus u_4u_6 \oplus u_2 \oplus u_5 \oplus u_3 \oplus u_4u_3,$$
$$u_2u_5 \oplus u_5u_1 \oplus u_1u_4 \oplus u_4 \oplus u_6 \oplus u_4u_5 \oplus u_2 \oplus u_3 \oplus u_3u_1,$$

$u_2u_5 \oplus u_5u_1 \oplus u_6u_5 \oplus u_1u_4 \oplus u_3u_5 \oplus u_1 \oplus u_4u_6 \oplus u_6u_3 \oplus u_4u_3 \oplus u_3u_1$,

$u_1u_4 \oplus u_2u_3 \oplus u_6u_1 \oplus u_4u_6 \oplus u_5 \oplus u_6u_3 \oplus u_4u_3$,

$u_5u_1 \oplus u_1u_4 \oplus u_6 \oplus u_3u_5 \oplus u_4u_5 \oplus u_1 \oplus u_6u_1 \oplus u_4u_6 \oplus u_3 \oplus u_6u_3 \oplus u_4u_2$,

$u_4 \oplus u_6 \oplus u_3u_5 \oplus u_1 \oplus u_4u_6 \oplus u_6u_3)$.

Pour calculer $\varphi(v_1 \oplus v_2)$, on utilise la fonction $\psi(x, y)=t(s(x)^4 \cdot s(y))$ de 12 bits sur 6 bits, qui donne les 6 bits de sortie en fonction des 12 bits d'entrée selon les règles suivantes :

$\psi(x_1, x_2, x_3, x_4, x_5, x_6, y_1, y_2, y_3, y_4, y_5, y_6) =$

$(x_3y_5 \oplus x_6y_2 \oplus x_6y_3 \oplus x_6y_4 \oplus x_3y_1 \oplus x_6y_1 \oplus x_1y_3 \oplus x_1y_5 \oplus x_5y_2 \oplus x_5y_5 \oplus x_5y_1 \oplus x_6y_6 \oplus x_1y_6 \oplus x_1y_2 \oplus x_1y_4 \oplus x_2y_1 \oplus x_2y_2 \oplus x_4y_4 \oplus x_3y_3 \oplus x_3y_6 \oplus x_4y_3 \oplus x_5y_3$,

$x_4y_5 \oplus x_3y_1 \oplus x_6y_1 \oplus x_2y_5 \oplus x_5y_1 \oplus x_6y_6 \oplus x_1y_6 \oplus x_1y_2 \oplus x_2y_1 \oplus x_2y_2 \oplus x_4y_1 \oplus x_4y_4 \oplus x_3y_3$,

$x_6y_2 \oplus x_6y_3 \oplus x_6y_4 \oplus x_6y_5 \oplus x_3y_1 \oplus x_6y_1 \oplus x_2y_5 \oplus x_5y_1 \oplus x_1y_6 \oplus x_1y_1 \oplus x_1y_2 \oplus x_1y_4 \oplus x_2y_1 \oplus x_2y_4 \oplus x_4y_2 \oplus x_2y_6 \oplus x_3y_4 \oplus x_5y_3$,

$x_3y_1 \oplus x_6y_2 \oplus x_2y_6 \oplus x_5y_3 \oplus x_5y_4 \oplus x_5y_6 \oplus x_6y_3 \oplus x_2y_3 \oplus x_4y_6 \oplus x_6y_5 \oplus x_1y_3 \oplus x_5y_5 \oplus x_2y_4 \oplus x_4y_2 \oplus x_4y_5 \oplus x_3y_5 \oplus x_4y_3 \oplus x_6y_1 \oplus x_4y_1$,

$x_3y_1 \oplus x_6y_6 \oplus x_5y_3 \oplus x_5y_6 \oplus x_5y_2 \oplus x_1y_5 \oplus x_1y_1 \oplus x_1y_2 \oplus x_2y_1 \oplus x_2y_3 \oplus x_3y_6 \oplus x_6y_5 \oplus x_1y_3 \oplus x_2y_4 \oplus x_3y_3 \oplus x_4y_5 \oplus x_2y_5 \oplus x_6y_1 \oplus x_4y_1 \oplus x_6y_4 \oplus x_3y_2$,

$x_6y_6 \oplus x_4y_4 \oplus x_5y_4 \oplus x_5y_6 \oplus x_6y_3 \oplus x_1y_6 \oplus x_1y_1 \oplus x_1y_2 \oplus x_2y_1 \oplus x_6y_5 \oplus x_2y_4 \oplus x_4y_2 \oplus x_4y_5 \oplus x_3y_5 \oplus x_6y_1 \oplus x_6y_4)$.

En utilisant ces formules, on calcule successivement :

- $\psi(v_1, v_1)$;
- $\psi(v_1, v_2)$;
- $\psi(v_2, v_1)$;
- $\psi(v_2, v_2)$.

Puis on calcule le "*OU-exclusif*" des quatre résultats obtenus.

Variante n°3 :

Toujours pour réduire la taille ROM nécessaire pour stocker les boîtes S, on peut enfin appliquer simultanément les idées des deux variantes précédentes, Variante n°1 et Variante n°2 : on utilise la Variante 2, avec la même bijection secrète φ (de 6 bits vers 6 bits) et la même fonction aléatoire secrète A (de 6 bits vers 6 bits) dans la nouvelle implémentation de chaque transformation non-linéaire donnée sous la forme d'une boîte-S.

Variante n°4 :

Dans cette dernière variante, au lieu de mettre en œuvre la transformation $(v'_1, v'_2)=S'(v_1, v_2)$, au moyen de deux boîtes S, qui remplace la transformation non-linéaire $v'=S(v)$ de la mise en œuvre initiale, qui était donnée sous la forme d'une boîte-S, on effectue le calcul de v'_1 respectivement v'_2 au moyen d'une fonction algébrique simple, pour laquelle les bits de v'_1 respectivement v'_2 sont donnés par une fonction polynomiale de degré total 1 ou 2 des bits de v_1 et v_2, puis on calcule v'_2 respectivement v'_1 au moyen d'une table. Cela permet encore de réduire la taille de la mémoire ROM nécessaire pour l'exécution matérielle.

Ainsi que représenté en figure 3e, dans le cas d'une étape de calcul du type transformation non linéaire mise en œuvre par un processus de calcul cryptographique classique, tel que le DES, outre la division en k parties v_1 à v_k d'une variable intermédiaire jouant le rôle d'entrée E, les transformations non linéaires consistent pour le processus classique, de même que dans le cas des figures 3b et 3d, en une transformation non linéaire de m bits sur n bits, décrite par des tables de conversion dans lesquelles les n bits de sortie de la transformation sont lus à une adresse fonction des m bits d'entrée, selon le procédé objet de l'invention, on remplace chaque transformation non linaire appliquée à une variable intermédiaire du processus de calcul cryptographique classique, en l'absence de division, par une transformation non linéaire partielle de km bits sur kn bits appliquée sur l'ensemble des variables intermédiaires partielles v_1 à v_k. Dans ce cas, et en référence à la Variante n°4 de l'exemple 2 précité, (k-1)n bits de sortie de cette transformation sont calculés comme une fonction polynomiale des km bits d'entrée des variables v_1, v_2 ... v_k selon les relations :

$$v'_1 = P_1(v_1, v_2, \ldots v_k)$$

.

.

.

$$v'_{k-1} = P_{k-1}(v_1, v_2, \ldots v_k),$$

relations dans lesquelles P_1 à P_{k-1} désignent des fonctions polynomiales de m bits sur n bits.

Les n bits restants v'_k de la variable de sortie sont alors obtenus, par exemple, par la lecture d'une table de

conversion non linéaire dans laquelle ces n bits sont lus à une adresse qui est fonction des km bits d'entrée.
Dans la Variante n°4 de l'exemple précité, on rappelle que k=2, m=6 et n=4.

Deuxième exemple : le Triple-DES

Le Triple-DES consiste à effectuer séquentiellement des opérations de chiffrement/déchiffrement à partir de deux clés secrètes.
Pour une description de l'algorithme Triple-DES, on pourra utilement se reporter à l'un des documents suivants :

- ISO/IEC 8732:1987, *Banking - Key Management (Wholesale);*
- ANSI X9.17, *American National Standard, Financial Institution Key Management (Wholesale)*, 1985.

ou encore dans l'ouvrage suivant :

- Bruce Schneier, *Applied Cryptography*, 2ème édition, John Wiley & Sons, 1996, page 358,

lesquels sont introduits dans la présente demande à titre de référence.

Le principe consiste à utiliser le DES trois fois de suite pour chiffrer un message : on commence par effectuer un DES en mode chiffrement avec la clé n°1, puis un DES en mode déchiffrement avec la clé n°2, et enfin un DES en mode chiffrement à nouveau avec la clé n°1. L'attaque de type DPA est possible de la même manière que pour le DES : grâce aux mesures de consommation effectuées sur le premier tour du premier DES, on trouve 48 bits de la clé n°1, puis en considérant le deuxième tour, on trouve les 8 bits restants de la clé n°1. Connaissant la clé n°1, on connaît alors les entrées du deuxième DES, et on peut appliquer la même attaque pour trouver la clé n°2.

La sécurisation de l'algorithme peut s'opérer exactement comme dans le cas du simple DES décrit dans le premier exemple ci-dessus : on utilise la même fonction *f* pour effectuer la "*division*" des variables intermédiaires, et les mêmes transformations de l'algorithme.

Troisième exemple : le RSA

Le RSA est le plus célèbre des algorithmes cryptographiques asymétriques. Il a été développé par Rivest, Shamir et Adleman en 1978. Pour une description plus détaillée de cet algorithme, on pourra utilement se reporter aux documents ci-après :

- R.L. Rivest, A. Shamir, L.M. Adleman, *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, 21, n°2, 1978, pp. 120-126,

ou dans les documents suivants :

- ISO/IEC 9594-8/ITU-T X.509, *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework;*
- ANSI X9.31-1, *American National Standard, Public-Key Cryptography Using Reversible Algorithms for the Financial Services Industry*, 1993;
- PKCS #1, *RSA Encryption Standard*, version 2, 1998, disponible à l'adresse suivante : ftp://ftp.rsa.com/pub/pkcs/doc/pkcs-1v2.doc.,

ces documents ou leur édition à titre de page HTML étant introduits dans la présente description à titre de référence.

L'algorithme RSA utilise un nombre entier *n* qui est le produit de deux grands nombres premiers *p* et *q*, et

un nombre entier e, premier avec $ppcm(p-1,q-1)$, et tel que $e \neq \pm 1 \bmod ppcm(p-1,q-1)$. Les entiers n et e constituent la clé publique. Le calcul en clé publique fait appel à la fonction g de Z/nZ dans Z/nZ définie par $g(x)=x^e \bmod n$. Le calcul en clé secrète fait appel à la fonction $g^{-1}(y)=y^d \bmod n$, où d est l'exposant secret (appelé aussi clé secrète, ou privée) défini par $ed \equiv 1 \bmod ppcm(p-1,q-1)$.

Les attaques de type DPA ou HO-DPA font également peser une menace sur les mises en œuvre classiques de l'algorithme RSA. En effet , celles-ci utilisent très souvent le principe dit de *square and multiply* en langage anglo-saxon pour effectuer le calcul de $x^d \bmod n$.

Ce principe consiste à écrire la décomposition $d = d_{m-1}\cdot 2^{m-1} + d_{m-2}\cdot 2^{m-2} + \ldots + d_1\cdot 2^1 + d_0\cdot 2^0$ de l'exposant secret d en base 2, puis d'effectuer le calcul de la manière suivante :

1. $z \leftarrow 1$;

   pour i allant de $m-1$ jusqu'à 0 faire :

2. $z \leftarrow z^2 \bmod n$;

3. si $d_i = 1$ alors $z \leftarrow z \times x \bmod n$.

Dans ce calcul, on constate que parmi les valeurs successives prises par la variable z, les premières ne dépendent que de quelques bits de la clé secrète d. L'hypothèse fondamentale permettant l'attaque DPA est donc réalisée. On peut ainsi deviner par exemple les 10 bits de poids fort de d en s'intéressant aux mesures de consommation sur la partie de l'algorithme correspondant à i allant de $m-1$ à $m-10$. On peut ensuite continuer l'attaque en utilisant les mesures de consommation sur la partie de l'algorithme correspondant à i allant de $m-11$ à

$m-20$, ce qui permet de trouver les 10 bits suivants de d, et ainsi de suite. On trouve finalement tous les bits de l'exposant secret d.

Le procédé objet de la présente invention s'applique également à la sécurisation de l'algorithme RSA. On utilise une division de chaque variable v intermédiaire, à valeurs dans le groupe multiplicatif de Z/nZ, c'est-à-dire dans l'ensemble des entiers modulo n qui possèdent un inverse, également modulo n, qui interviennent dans le cours du calcul et qui dépendent des données d'entrée ou de sortie, en deux variables v_1 et v_2. On prend $k=2$ et on prend la fonction $f(v_1, v_2) = v = v_1 \cdot v_2 \mod n$. On a déjà montré précédemment dans la description, confer exemple n°2 *"Sécurisation de l'algorithme"*, que cette fonction f permet de satisfaire la condition n°1.

On remplace donc x par (x_1, x_2) tel que $x = x_1 \cdot x_2 \mod n$ et z par (z_1, z_2) tel que $z = z_1 \cdot z_2 \mod n$. En pratique, on peut par exemple choisir x_1 aléatoirement et en déduire x_2. En reprenant maintenant les trois étapes de la méthode "*square and multiply*", on effectue les transformations suivantes :

1.« $z \leftarrow 1$ » est remplacé par « $z_1 \leftarrow 1$ et $z_2 \leftarrow 1$ »;

2.« $z \leftarrow z^2 \mod n$ » est remplacé par « $z_1 \leftarrow z_1^2 \mod n$ et $z_2 \leftarrow z_2^2 \mod n$ » ;

3.« $z \leftarrow z \times x \mod n$ » est remplacé par . « $z_1 \leftarrow z_1 \times x_1 \mod n$ et $z_2 \leftarrow z_2 \times x_2 \mod n$ ».

Il est facile de vérifier que la relation $z=f(z_1, z_2)$ reste vraie tout au long du calcul, ce qui montre que la condition n°2 est satisfaite.

On remarquera que les calculs effectués respectivement sur la variable z_1 et sur la variable z_2

sont complètement indépendants. On peut donc prévoir d'exécuter les deux calculs :

- soit séquentiellement ;
- soit de façon imbriquée ;
- soit de façon simultanée dans le cas de la multiprogrammation ;
- soit encore simultanément dans des processeurs différents travaillant en parallèle.

REVENDICATIONS

1. Procédé de sécurisation d'un ensemble électronique mettant en œuvre un processus de calcul cryptographique classique qui utilise une clé secrète, caractérisé en ce que :

a) l'on divise le processus de calcul cryptographique classique en plusieurs parties de processus de calcul distinctes conduites parallèlement et mettant en œuvre des résultats partiels intermédiaires distincts de ceux du calcul cryptographique classique ;

b) on reconstitue la valeur finale obtenue par le calcul cryptographique classique en l'absence de division, à partir desdits résultats partiels intermédiaires distincts.

2. Procédé selon la revendication 1, caractérisé en ce que chaque variable ou résultat (v) intermédiaire dépendant des données d'entrée ou de sortie mises en œuvre par le processus de calcul cryptographique classique est remplacé par un nombre déterminé k de variables intermédiaires partielles (v_1, ..., v_k), les variables intermédiaire (v) et intermédiaires partielles (v_1 à v_k) étant liées par une fonction f, $v=f(v_1, v_2, ..., v_k)$ permettant la reconstitution de ladite variable intermédiaire (v).

3. Procédé selon la revendication 2, caractérisé en ce que ladite fonction f, liant les variables intermédiaires partielles et ladite variable intermédiaire (v), est telle que la connaissance d'une valeur de cette variable intermédiaire ne permet jamais de déduire l'ensemble des valeurs particulières partielles v_i telles

qu'il existe un *(k-1)-uplet* $(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots v_k)$ satisfaisant à l'équation $f(v_1, \ldots, v_i, \ldots, v_k) = v$.

4. Procédé selon la revendication 3, caractérisé en ce que ladite fonction est la fonction *"OU-exclusif"* bit à bit, lesdites variable intermédiaire (v) et variables intermédiaires partielles $(v_1, \ldots, v_i, \ldots, v_k)$ vérifiant la relation :

$$f(v_1, \ldots, v_i, \ldots, v_k) = v_1 \oplus v_2 \oplus, \ldots \oplus v_i \oplus v_k.$$

5. Procédé selon la revendication 3, caractérisé en ce que pour une variable intermédiaire (v) à valeurs dans le groupe multiplicatif Z/nZ défini par l'ensemble des entiers modulo n, ladite fonction est la fonction produit modulo n, $f(v_1,\ldots,v_k)=v_1.v_2\ldots v_k$ modulo n, dans laquelle lesdites variables intermédiaires partielles sont des variables à valeurs dans ledit groupe multiplicatif de Z/nZ.

6. Procédé selon la revendication 3, caractérisé en ce que, ladite fonction f liant les variables intermédiaires partielles et ladite variable intermédiaire (v), les parties de processus de calcul distinctes conduites parallèlement sont indépendantes, lesdites parties de processus de calcul distinctes conduites parallèlement étant conduites en l'absence de reconstitution de ladite variable intermédiaire (v) dépendant des données d'entrée ou de sortie mise en œuvre par ledit processus de calcul cryptographique classique.

7. Procédé selon la revendication 1, caractérisé en ce que ladite division est effectuée en deux parties distinctes conduites parallèlement.

8. Procédé selon la revendication 1, caractérisé en ce que ladite division est effectuée en k parties, et en ce que, pour un processus de calcul cryptographique classique utilisant des transformations non linéaires de *m* bits sur *n* bits décrites par des tables de conversion dans lesquelles les *n* bits de sortie de la transformation sont lus à une adresse fonction des *m* bits d'entrée, on remplace chaque transformation non linéaire appliquée à une variable intermédiaire du processus de calcul cryptographique classique, en l'absence de division, par une transformation non linéaire partielle de *km* bits sur *kn* bits appliquée à l'ensemble des variables intermédiaires partielles, cette transformation non linéaire partielle étant décrite par *k* tables de conversion partielle dans lesquelles les *n* bits de sortie de la transformation sont lus à une adresse fonction des *km* bits d'entrée.

9. Procédé selon la revendication 8, caractérisé en ce que parmi les *k* tables de conversion partielle *k-1* tables de conversion partielle contiennent des variables aléatoires secrètes.

10. Procédé selon la revendication 8, caractérisé en ce que parmi les *k* tables de conversion partielle, utilisées pour remplacer chaque table de conversion non linéaire, on utilise à chaque fois les mêmes *k-1* tables aléatoires secrètes.

11. Procédé selon la revendication 1, caractérisé en ce que ladite division est effectuée en k parties, et en ce que, pour un processus de calcul cryptographique classique utilisant des transformations non linéaires de *m* bits sur *n* bits décrites par des tables de conversion dans

lesquelles les *n* bits de sortie de la transformation sont lus à une adresse fonction des *m* bits d'entrée, on remplace chaque transformation non linéaire appliquée à une variable intermédiaire du processus de calcul cryptographique classique, en l'absence de division, par une transformation non linéaire partielle de *km* bits sur *kn* bits appliquée sur l'ensemble des variables intermédiaires partielles, *(k-1)n* desdits bits de sortie de cette transformation étant calculés comme fonction polynomiale des *km* bits d'entrée et les *n* bits restants desdits bits de sortie étant obtenus par lecture d'une table de conversion dans laquelle les *n* bits restants sont lus à une adresse qui est fonction des *km* bits d'entrée.

12. Procédé selon la revendication 2, caractérisé en ce que ladite division est effectuée en k parties, et en ce que, pour un processus de calcul cryptographique classique utilisant des transformations non linéaires de *m* bits sur *n* bits décrites par des tables de conversion dans lesquelles les *n* bits de sortie de la transformation sont lus à une adresse fonction des *m* bits d'entrée, on remplace chaque transformation non linéaire appliquée à une variable intermédiaire du processus de calcul cryptographique classique, en l'absence de division, par une transformation non linéaire partielle de *km* bits sur *kn* bits appliquée sur l'ensemble des variables intermédiaires partielles, cette transformation non linéaire partielle étant décrite par k tables de conversion, chacune de ces tables de conversion recevant comme entrée une valeur obtenue par application d'une fonction bijective secrète φ_1 à ladite fonction $f(v_1,\ldots, v_k)$ des variables intermédiaires partielles selon la relation $\varphi_j \circ f(v_1, \ldots, v_k)$, $j \in [1,k]$, cette application $\varphi_j \circ f(v_1,\ldots,$

v_k) étant effectuée par évaluation directe d'une valeur résultante, cette valeur résultante, appliquée à l'entrée de la table de conversion, permettant de lire *n* bits de sortie de la transformation à une adresse qui est fonction de ces *m* bits d'entrée.

13. Procédé selon la revendication 12, caractérisé en ce que, parmi les k tables de conversion partielle, k-1 tables de conversion partielle contiennent des valeurs aléatoires secrètes.

14. Procédé selon la revendication 12, caractérisé en ce que, parmi les k tables de conversion partielle utilisées pour remplacer chaque table de transformation non linéaire, on utilise à chaque fois les mêmes k-1 tables de conversion aléatoires secrètes.

15. Procédé selon la revendication 1, caractérisé en ce que les opérations effectuées dans les différentes parties issues de la division du processus de calcul cryptographique en plusieurs parties de processus de calcul distinctes sont exécutées séquentiellement.

16. Procédé selon la revendication 1, caractérisé en ce que les opérations effectuées dans les différentes parties issues de la division du processus de calcul cryptographique en plusieurs parties de processus de calcul distinctes sont exécutées de façon imbriquée.

17. Procédé selon la revendication 1, caractérisé en ce que les opérations effectuées dans les différentes parties issues de la division du processus de calcul cryptographique en plusieurs parties de processus de calcul distinctes sont exécutées de façon simultanée dans le cas de la multiprogrammation.

18. Procédé selon la revendication 1, caractérisé en ce que les opérations effectuées dans les différentes parties issues de la division du processus de calcul cryptographique en plusieurs parties de processus de calcul distinctes sont exécutées simultanément dans des processeurs différents travaillant en parallèle.

19. Utilisation du procédé selon la revendication 1 dans une carte à microcalculateur.

20. Utilisation du procédé selon la revendication 1 pour la sécurisation de processus de calcul cryptographique supporté par les algorithmes DES, Triple DES, RSA.

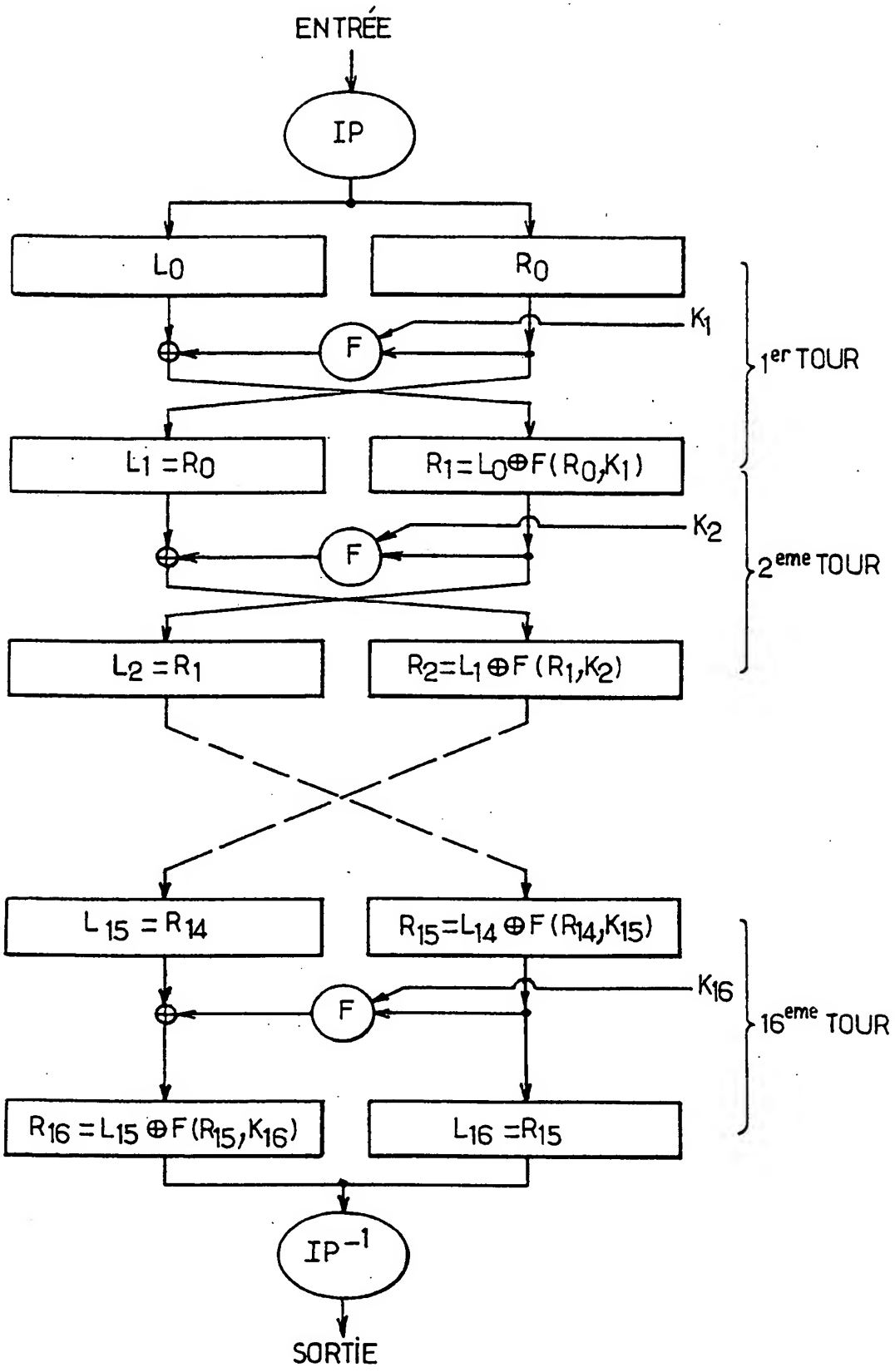


FIG.1a.

L'ALGORITHME DES (ART ANTÉRIEUR)

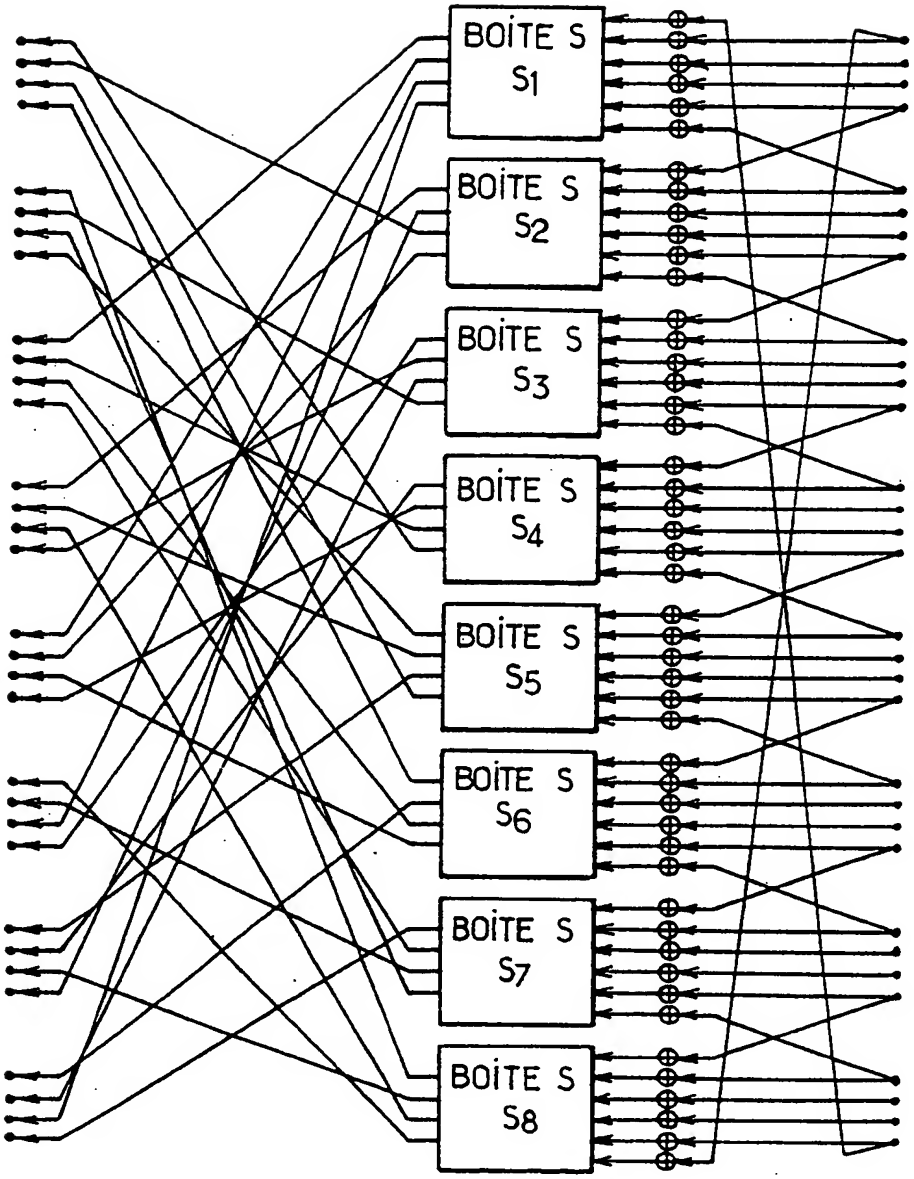
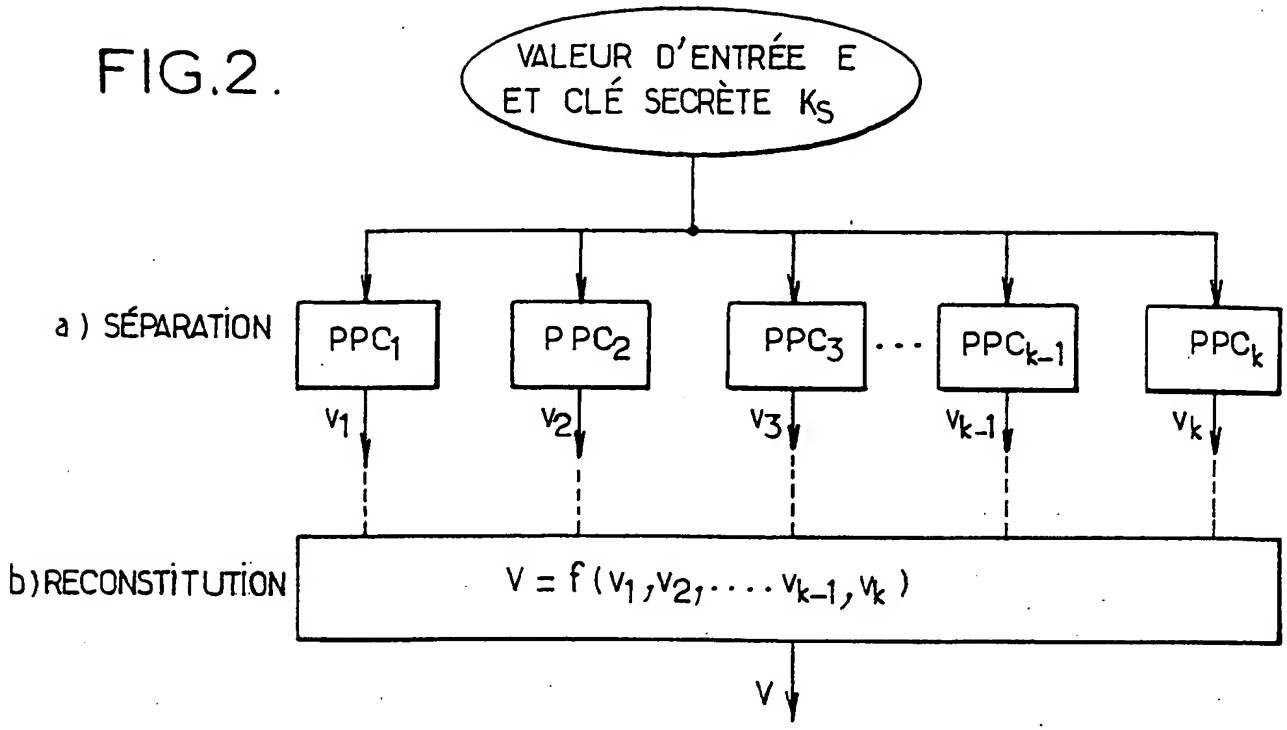


FIG.1b.

LA FONCTION F DU DES (ART ANTÉRIEUR)

FIG.2.


VALEUR D'ENTRÉE E
ET CLÉ SECRÈTE Ks
a) SÉPARATION
PPC1
PPC2
PPC3
PPCk-1
PPCk
v1
v2
v3
vk-1
vk
b) RECONSTITUTION
V = f (v1, v2, ... vk-1, vk)
V

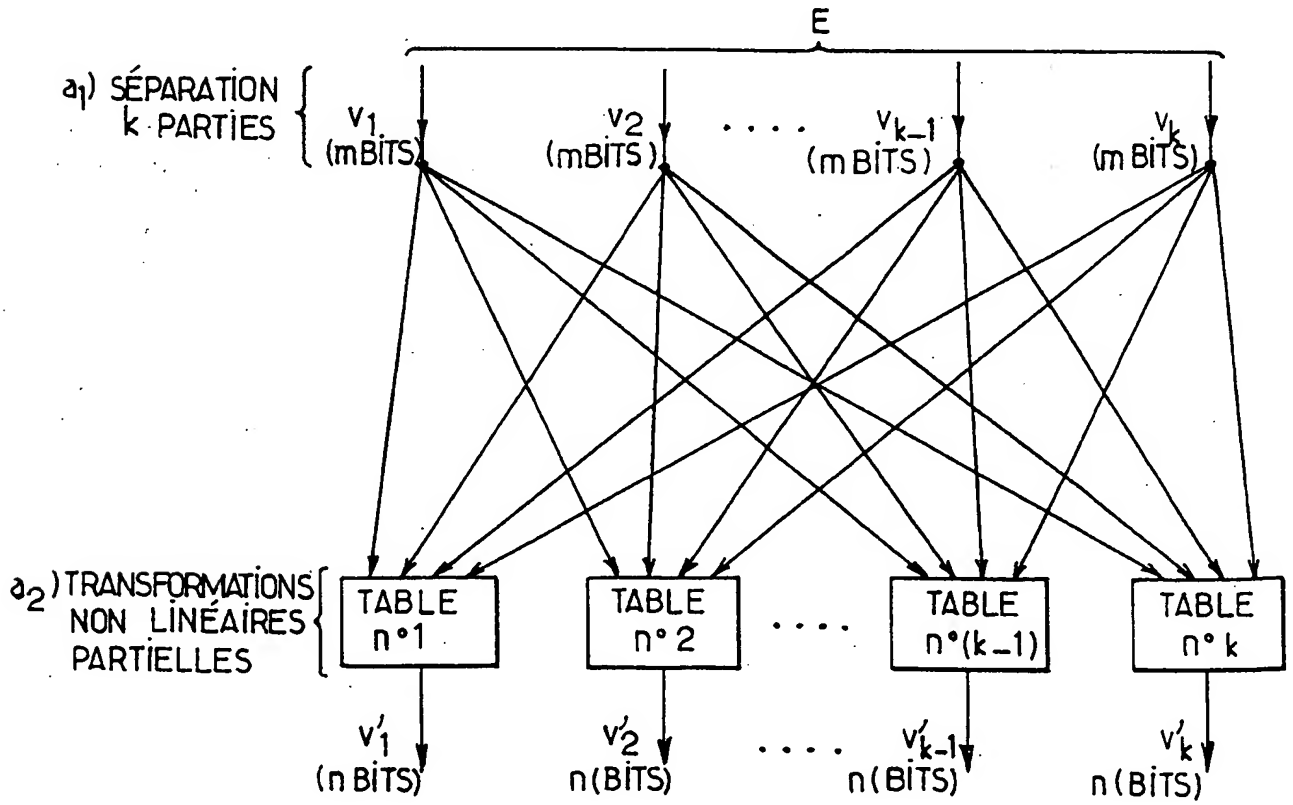
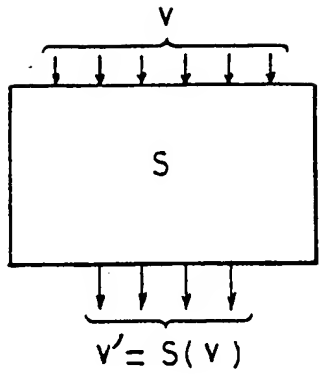
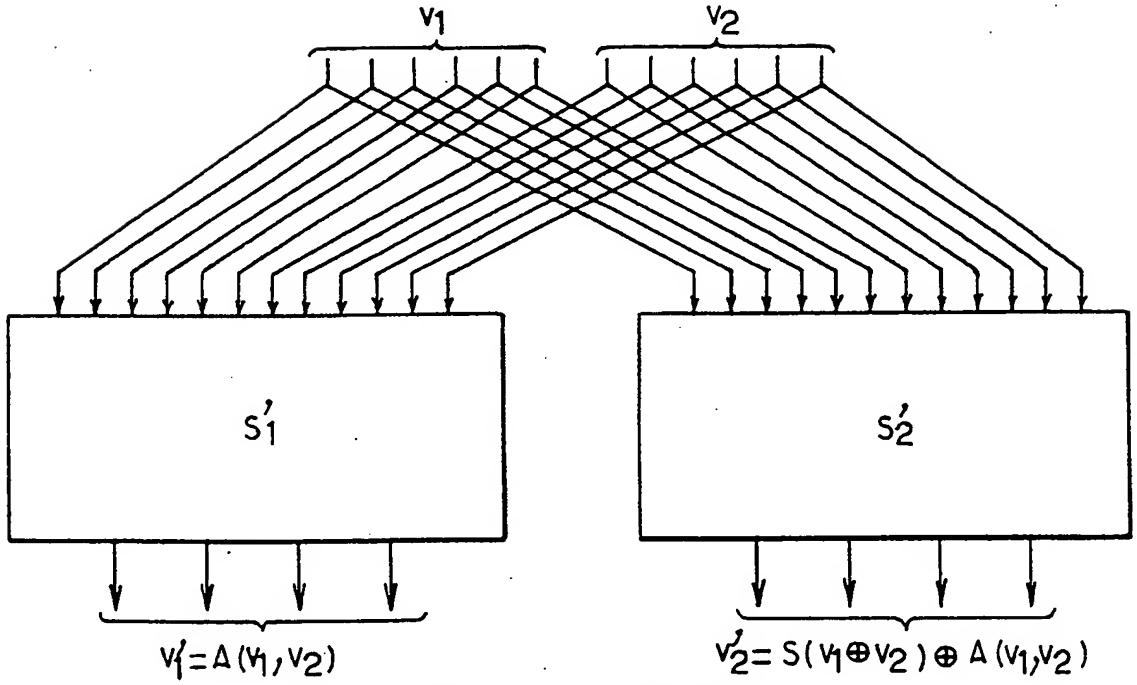

E
a1) SÉPARATION
k PARTIES
v1
(mBITS)
v2
(mBITS)
vk-1
(mBITS)
vk
(mBITS)
a2) TRANSFORMATIONS
NON LINÉAIRES
PARTIELLES
TABLE
n°1
TABLE
n° 2
TABLE
n°(k-1)
TABLE
n° k
v'1
(n BITS)
v'2
n(BITS)
v'k-1
n(BITS)
v'k
n(BITS)

FIG.3b.



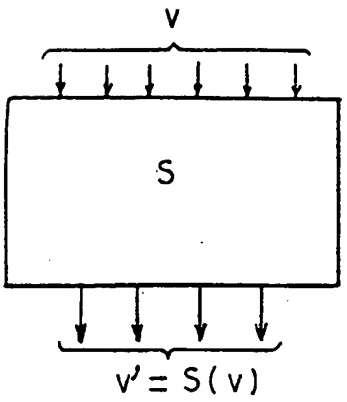
PROCESSUS CLASSIQUE



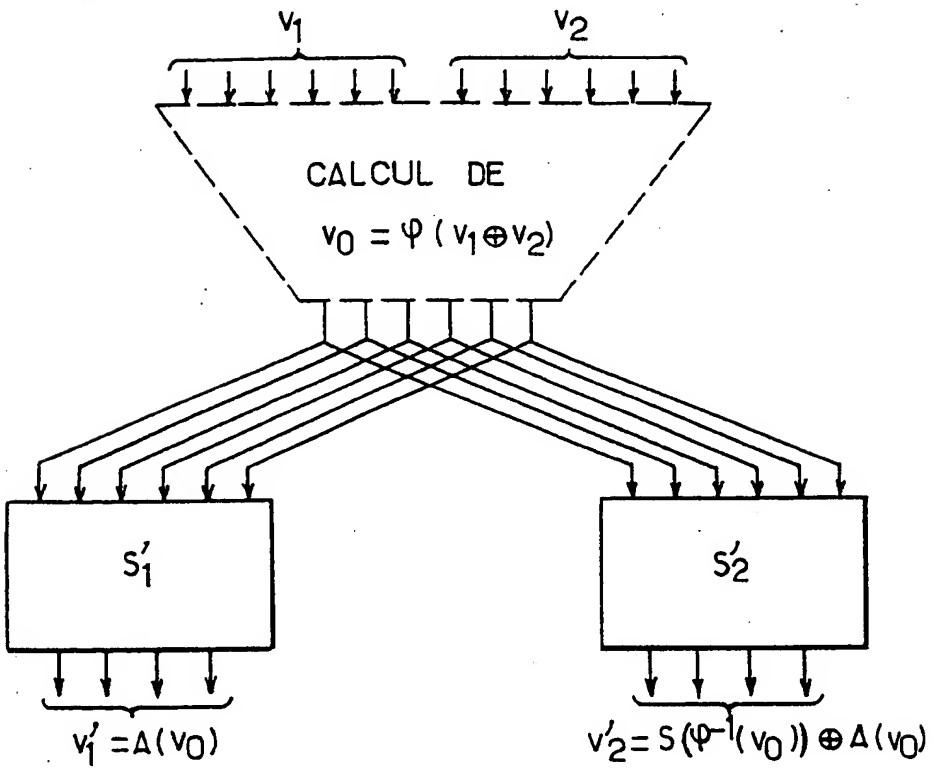
PROCESSUS MODIFIÉ

FIG. 3a.

TRANSFORMATION D'UNE BOITE S



PROCESSUS CLASSIQUE



PROCESSUS MODIFIÉ

FIG.3c.

TRANSFORMATION D'UNE BOITE S (VARIANTE N°2)

FIG.3d.

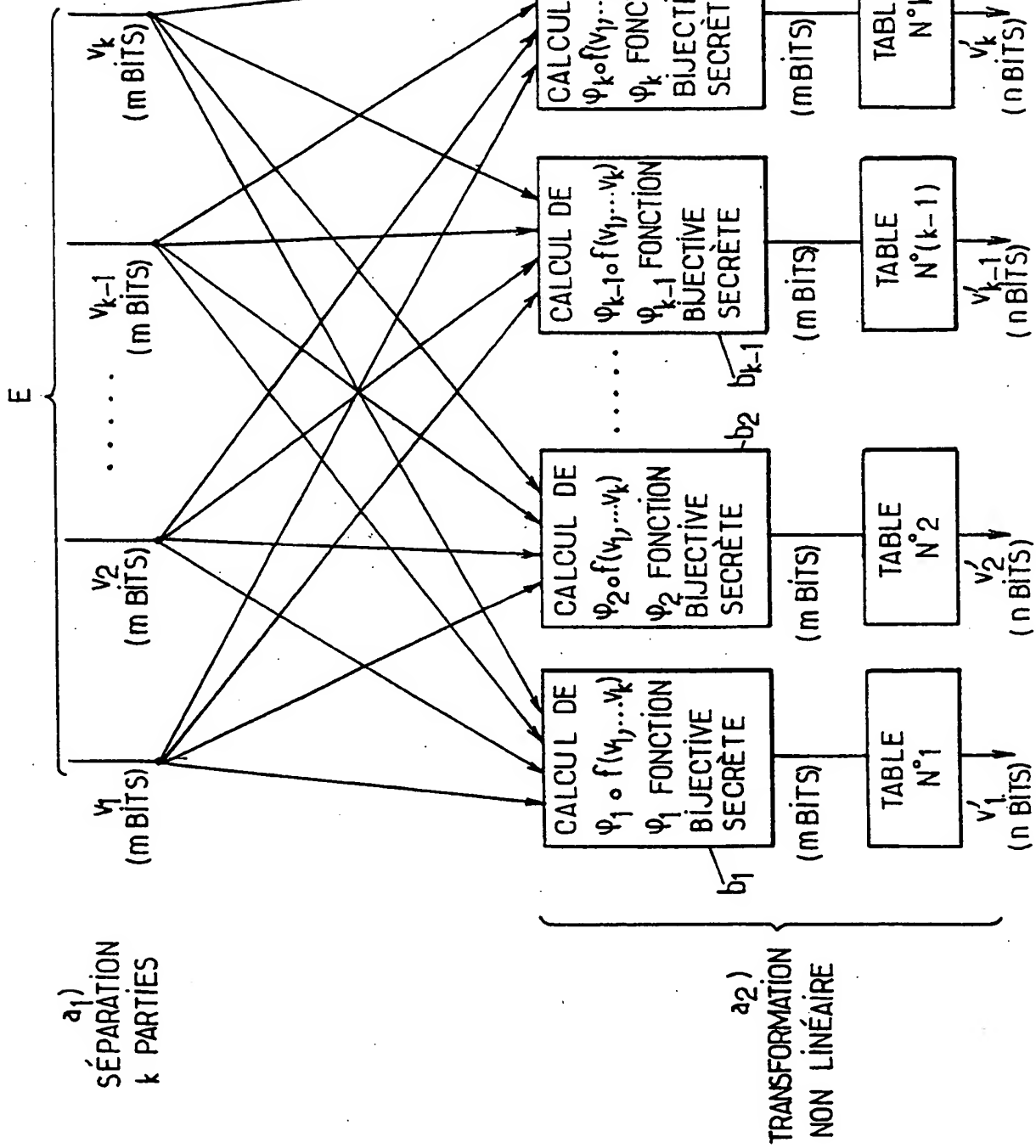
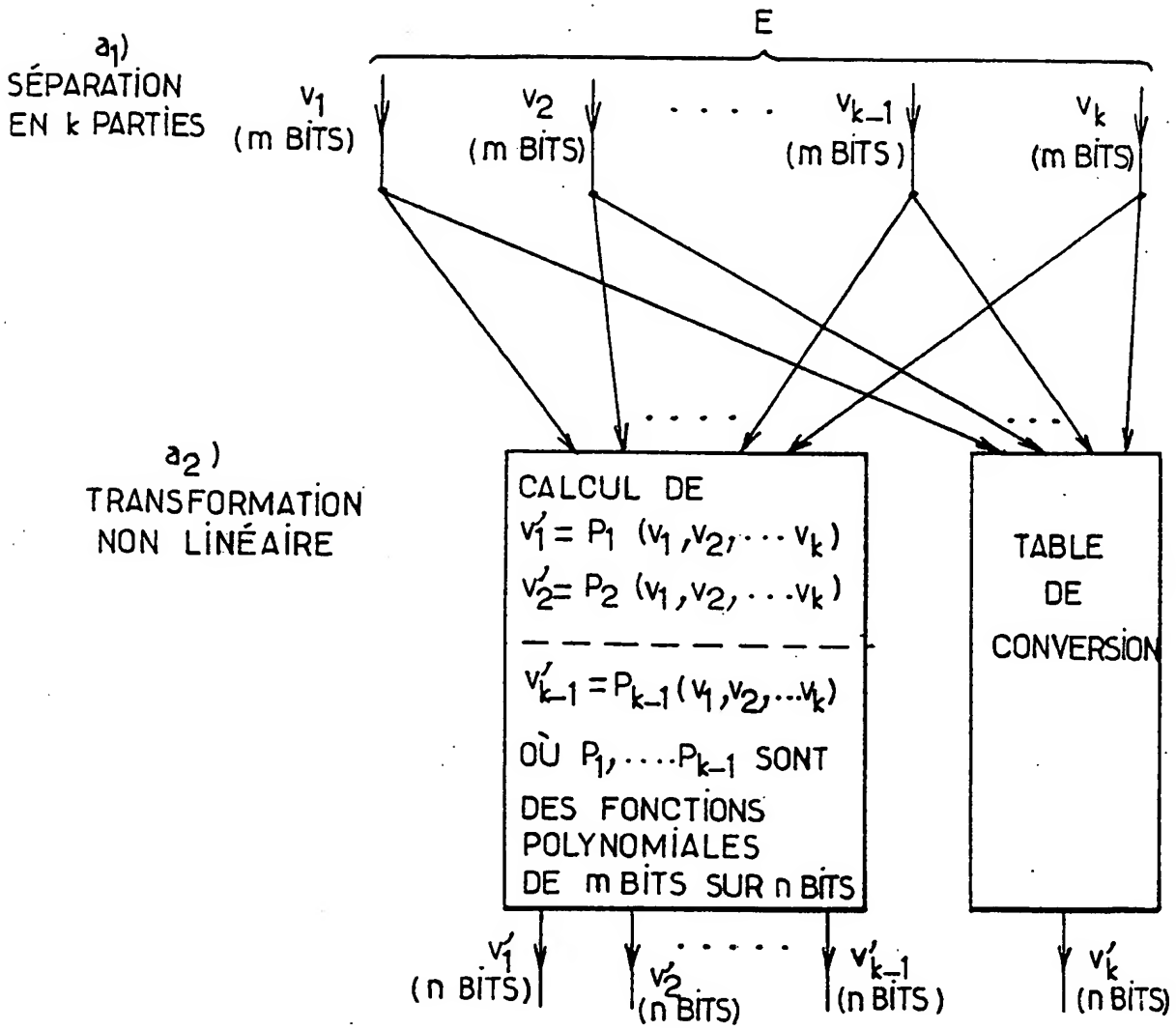

E
v_1 (m BITS)
v_2 (m BITS)
v_{k-1} (m BITS)
v_k (m BITS)
a_1) SÉPARATION k PARTIES
a_2) TRANSFORMATION NON LINÉAIRE
CALCUL DE $\varphi_1 \circ f(v_1,...v_k)$ φ_1 FONCTION BIJECTIVE SECRÈTE
CALCUL DE $\varphi_2 \circ f(v_1,...v_k)$ φ_2 FONCTION BIJECTIVE SECRÈTE
CALCUL DE $\varphi_{k-1} \circ f(v_1,...v_k)$ φ_{k-1} FONCTION BIJECTIVE SECRÈTE
CALCUL DE $\varphi_k \circ f(v_1,...v_k)$ φ_k FONCTION BIJECTIVE SECRÈTE
b_1
b_2
b_{k-1}
b_k
(m BITS)
TABLE N°1
TABLE N°2
TABLE N°(k-1)
TABLE N°k
v'_1 (n BITS)
v'_2 (n BITS)
v'_{k-1} (n BITS)
v'_k (n BITS)

FIG.3e.


E
a1)
SÉPARATION
EN k PARTIES
v1
(m BITS)
v2
(m BITS)
vk−1
(m BITS)
vk
(m BITS)
a2)
TRANSFORMATION
NON LINÉAIRE
CALCUL DE
v'1 = P1 (v1, v2, ... vk)
v'2 = P2 (v1, v2, ... vk)
v'k−1 = Pk−1 (v1, v2, ... vk)
OÙ P1, ... Pk−1 SONT
DES FONCTIONS
POLYNOMIALES
DE m BITS SUR n BITS
TABLE
DE
CONVERSION
v'1
(n BITS)
v'2
(n BITS)
v'k−1
(n BITS)
v'k
(n BITS)

INTERNATIONAL SEARCH REPORT

Intern nal Application No
PCT/FR 00/00258

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/06 H04L9/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 98 52319 A (YEDA RES & DEV ;FLEIT LOIS (US)) 19 November 1998 (1998-11-19)	1,20
A	page 12, line 6 - last line abstract	6,7

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
25 April 2000	03/05/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

1

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern ıal Application No
PCT/FR 00/00258

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9852319 A	19-11-1998	US 5991415 A	23-11-1999
		AU 7568598 A	08-12-1998
		EP 0986873 A	22-03-2000

RAPPORT DE RECHERCHE INTERNATIONALE

Dem. internationale No
PCT/FR 00/00258

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 H04L9/06 H04L9/30

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 H04L

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de donnée, et si réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie *	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
X	WO 98 52319 A (YEDA RES & DEV ;FLEIT LOIS (US)) 19 novembre 1998 (1998-11-19)	1,20
A	page 12, ligne 6 - dernière ligne abrégé	6,7

☐ Voir la suite du cadre C pour la fin de la liste des documents

☒ Les documents de familles de brevets sont indiqués en annexe

* Catégories spéciales de documents citées:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
25 avril 2000	03/05/2000

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Holper, G

1

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Dema Internationale No
PCT/FR 00/00258

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
WO 9852319 A	19-11-1998	US 5991415 A	23-11-1999
		AU 7568598 A	08-12-1998
		EP 0986873 A	22-03-2000